

# Descripción de la BD para las prácticas

## Enunciado general

“Restaurante en casa S.A.” es una compañía que distribuye pedidos de comida desde restaurantes a casas particulares y oficinas. La compañía ha decidido utilizar una aplicación para su negocio que soporte los siguientes tipos de usuarios:

- Clientes: realizan pedidos y consultan su estado.
- Atención a cliente: atienden llamadas de clientes (realización de pedidos y consulta de estado).
- Restaurantes: mantenimiento de menús y preparación de pedidos.
- Distribuidores: asignan repartidor a cada pedido.
- Repartidores: recogen pedidos de restaurantes y los entregan en destino.

El escenario principal de la aplicación corresponde al flujo de un pedido. La secuencia es la siguiente: (1) el cliente realiza el pedido a través del interfaz web; (2) el sistema envía el pedido al restaurante correspondiente vía fax ó email; (3) el restaurante confirma que ha recibido el pedido; (4) un distribuidor asigna el pedido a un repartidor; (5) el sistema envía la notificación al repartidor correspondiente; (6) el repartidor ve el pedido asignado en su teléfono móvil; (7) el repartidor recoge el pedido del restaurante y notifica la recogida al sistema; (8) el repartidor entrega el pedido y confirma al sistema la entrega. Casos de uso a considerar:

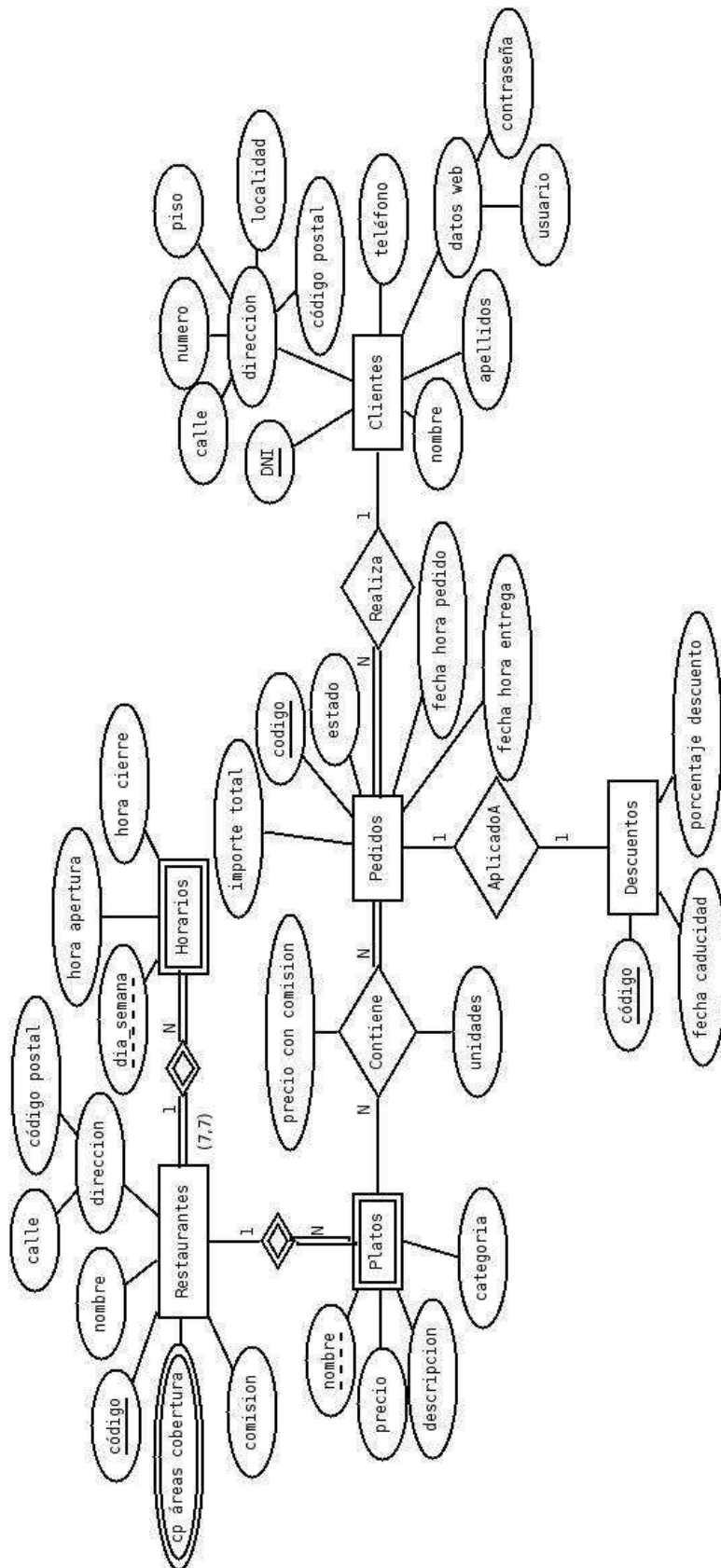
- Realizar pedido: el cliente realiza un pedido a través del interfaz web
- Ver pedidos: el usuario de atención al cliente puede ver los pedidos
- Enviar pedidos a restaurante: el sistema envía el pedido
- Confirmación de recepción de pedido
- Modificación de pedido

## Descripción de la base de datos

La aplicación de “Restaurante en casa S.A.” debe almacenar la siguiente información:

- De cada restaurante se conoce su nombre, dirección, horario detallado para cada día de la semana y áreas de cobertura, que serán las localidades a las que se pueden servir pedidos desde el restaurante. Para facilitar la gestión de datos, cada restaurante tiene un código identificativo único.
- Cada restaurante ofrece un conjunto de platos distintos que se pueden elegir de manera individual en cada pedido. Cada plato tiene un nombre, descripción y un precio, al que se debe agregar la comisión que la compañía aplica a cada restaurante. En el sistema, los platos se agrupan en categorías comunes a todos los restaurantes (e.g. pescados, arroces, etc.).
- Los clientes de la empresa facilitan su DNI, nombre, apellidos, dirección (calle, número, piso, localidad, código postal) y un número de teléfono de contacto la primera vez que utilizan el servicio de “Restaurante en casa S.A.”, de modo que quedan registrados en el sistema. Para consultar el estado de sus pedidos, cada cliente puede disponer de un usuario y una contraseña.
- Los pedidos que cada cliente realiza se componen de la siguiente información: código de pedido (automáticamente generado a partir de una secuencia que se inicia a 1), fecha del pedido, fecha de entrega, estado del pedido, importe total y cliente. Interesa además registrar los platos que componen el pedido, indicando las unidades y precio de cada uno de ellos (comisión incluida).
- La compañía distribuye cupones descuento cuyos datos son su código (no se repite), fecha de caducidad y porcentaje de descuento. Al hacer un pedido, el cliente indicará, si dispone de alguno, el código de su cupón. En ese caso, el pedido debe incluir la información de descuento, aplicándolo, en consecuencia, al importe final del pedido.

El Modelo E/R de la BD queda:



# Práctica 1: Toma de contacto con el SGBD Oracle y consultas SQL

Entrega 7 de Noviembre

## Objetivos

- Creación de usuarios y configuración del espacio de trabajo
- Manejo del entorno Oracle SQL Developer. Comandos básicos.
- Creación y manejo de tablas con SQL.
- Consultas en SQL para ejemplificar la sintaxis concreta de SQL soportado por ORACLE.

## Evaluación

Cada grupo subirá al campus virtual un informe con sus nombres e incluya de forma razonada todas las sentencias que se han ejecutado. Se puede incluir:

- Ficheros .sql con scripts de DDL, y otro con SQL DML (para apartado 4 de consultas)
- Un documento explicativo en Word para cada paso
- Si es por parejas, un fichero de texto con los dos nombres completos

## Introducción

Se puede usar la interfaz gráfica Oracle SQL Developer.

Si se usa la primera (recomendada) se deben importar las conexiones definidas previamente (si no aparecen). Para ello se debe iniciar Oracle SQL Developer (en Inicio/software/bbdd/Oracle). Pulsar con el botón secundario del ratón el icono Conexiones. Seleccionar Importar conexiones y escoger el fichero **c:\software\bbdd\oracle\sqldeveloper\Conexiones**. La conexión que debemos escoger es **BDiC**. Más adelante se usará un usuario para empezar a trabajar y se solicitará tanto su identificador como su contraseña.

## Espacio de trabajo

Como paso previo a la realización de las prácticas es necesario que cada grupo de prácticas prepare su espacio de trabajo, es decir, que cree su TABLESPACE y su usuario.

1. Iniciar sesión en PL/SQL con el usuario *ADMINUSER* (contraseña *ADMINUSER*).
2. Crear un espacio para tablas *EMPRESAGIICxx*
  - *xx* es el número de grupo (Laboratorio 1: número de puesto, laboratorio 2: número de puesto + 20)
  - *Ruta* es la ruta al directorio de datos de la instancia de Oracle (**D:\oracle**).

```
CREATE TABLESPACE EMPRESAGIICxx DATAFILE 'D:\oracle\EMPRESAGIICxx'
SIZE 5M AUTOEXTEND OFF;
```

3. Crear un usuario *GIICxx* :
  - *XX* es el número de grupo (de 01 a 40)

```
CREATE USER GIICxx IDENTIFIED BY GIICxx DEFAULT TABLESPACE
EMPRESAGIICxx TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON
EMPRESAGIICxx;
```

4. Asignar permisos al usuario:

```
GRANT CREATE SESSION, CREATE TABLE, DELETE ANY TABLE, SELECT ANY
DICTIONARY, CREATE ANY SEQUENCE TO GIICxx;
```

5. Cerrar la sesión del usuario ADMINUSER.

Iniciar sesión con el nuevo usuario GIICxx.

## Apartado 1: Crear y ejecutar el script crea\_tablas.sql

Codifica las sentencias de creación de las tablas de la base de datos. Usa los identificadores de los objetos Oracle (tablas, campos, etc.) exactamente como se indica en el enunciado. Consúltase el prontuario de Oracle proporcionado en la página web para saber cómo usar identificadores con caracteres especiales (e.g., espacios).

```
Restaurantes (codigo Number(8), nombre Char(20), calle Char(30)
              , "codigo postal" Char(5), comision* Number(8, 2))
Áreas Cobertura (restaurante Number(8), "codigo postal" Char(5))
Horarios (restaurante Number(8), "dia semana" Char(1)
          , hora_apertura Date, hora_cierre Date)
Platos (restaurante Number(8), nombre Char(20), precio* Number(8,2)
        , descripción* Char(30), categoría* Char(20))
Pedidos (codigo Number(8), estado Char(9), fecha_hora_pedido Date
         , fecha_hora_entrega* Date, "importe total"* Number(8,2)
         , cliente Char(9), codigoDescuento* Number(8))
Contiene (restaurante Number(8), plato Char(20), pedido Number(8)
         , "precio con comisión"* Number(8,2), unidades Number(4))
Descuentos (codigo Number(8), fecha_caducidad* Date
            , "porcentaje descuento" Number(3))
Clientes (DNI Char(9), nombre, apellido, calle* Char(20)
          , numero Number(4), piso* Char(5), localidad* Char(15)
          , "codigo postal"* Char(5), teléfono* Char(9)
          , usuario Char(8), contraseña Char(8))
```

Incluye las restricciones adicionales que consideres adecuadas teniendo en cuenta las siguientes consideraciones:

- El descuento de los cupones debe ser mayor que cero y menor o igual a cien.
- Se necesita un índice que optimice la búsqueda de platos según su categoría.
- Existen clientes que no han personalizado su contraseña porque no han accedido nunca por internet (sus pedidos son telefónicos). En ese caso, la contraseña por defecto es “Nopass”. El nombre de usuario es obligatorio, debiendo verificar que no se repite.
- El estado por defecto de todos los pedidos es “REST”, y debe ser “REST”, “CANCEL”, “RUTA”, “ENTREGADO” ó “RECHAZADO”.
- Para las claves ajenas configurar el comportamiento de la clave ajena cuando se eliminan filas referenciadas. Por ejemplo:
  - Si se borra un descuento se debe poner automáticamente el campo codigoDescuento a NULL en su pedido correspondiente
  - Si se borra un restaurante se eliminan automáticamente sus platos y sus horarios
  - Si se borra un pedido se debe eliminar automáticamente los platos que contiene
  - No se puede eliminar un cliente mientras tenga pedidos
  - No se puede borrar un plato mientras existan pedidos con dicho plato

**Notas:** Si un identificador contiene un espacio, se debe encerrar entre comillas dobles (").

Se recomienda escribir, ejecutar y comprobar el resultado de la ejecución de las consultas, una a una, no todas a la vez.

**Ejemplos:**

```
SELECT TABLE_NAME FROM USER_TABLES;  
-- Permite ver las tablas creadas por el usuario conectado
```

```
CREATE TABLE Restaurantes(  
    codigo NUMBER(8) NOT NULL  
    , nombre CHAR(20) NOT NULL  
    , calle CHAR(30) NOT NULL  
    , "código postal" CHAR(5) NOT NULL  
    , comision NUMBER(8,2)  
    , PRIMARY KEY(codigo)  
);
```

```
CREATE INDEX I_CatPlatos ON Platos(categoria);
```

```
CREATE SEQUENCE Seq_CodPedidos INCREMENT BY 1 START WITH 1  
NOMAXVALUE;
```

```
CREATE TABLE Pedidos(  
    codigo NUMBER(8) NOT NULL  
    , estado CHAR(9) DEFAULT 'REST' NOT NULL  
    , fecha_hora_pedido DATE NOT NULL  
    , fecha_hora_entrega DATE  
    , "importe total" NUMBER(8,2)  
    , cliente CHAR(9) NOT NULL REFERENCES Clientes(DNI)  
    , codigoDescuento Number(8) REFERENCES Descuentos(codigo) ON  
DELETE SET NULL  
    , PRIMARY KEY(codigo)  
    , CHECK (estado IN ('REST', 'CANCEL', 'RUTA', 'ENTREGADO',  
'RECHAZADO'))  
);
```

```
CREATE TABLE Contiene(  
    restaurante NUMBER(8)  
    , plato CHAR(20)  
    , pedido NUMBER(8) REFERENCES Pedidos(codigo) ON DELETE  
CASCADE  
    , "precio con comisión" NUMBER(8,2)  
    , unidades NUMBER(4) NOT NULL  
    , PRIMARY KEY(restaurante, plato, pedido)  
    , FOREIGN KEY(restaurante, plato) REFERENCES  
Platos(restaurante, nombre)  
);
```

```
CREATE TABLE Descuentos(  
    codigo NUMBER(8) NOT NULL  
    , fecha_caducidad DATE  
    , "porcentaje descuento" NUMBER(3) CHECK ("porcentaje  
descuento" >0 AND "porcentaje descuento" <=100)  
    , PRIMARY KEY(codigo));
```

## Apartado 2: Carga de datos con INSERT

Insertar al menos una fila en cada tabla, y añadir en el informe el código de error si:

- Se intenta insertar una fila con clave primaria duplicada
- Se intenta una inserción que no incluye todas las columnas no nulas. Ejemplo:  
`INSERT INTO Contiene (restaurante) VALUES (1234);`
- Se intenta una inserción que no verifica las restricciones de dominio (CHECK)
- Se intenta una inserción que no respeta una regla de integridad referencial
- Se intenta un borrado en una tabla padre con filas dependientes con una regla de borrado ON DELETE RESTRICT (el defecto), y documentar que pasa si se borra en una tabla padre con filas dependientes con regla de borrado ON DELETE SET NULL y ON DELETE RESTRICT.

Nota para borrar una fila: `DELETE FROM tabla WHERE PK='valor'`

Algunos ejemplos de inserción (que deben hacerse en el orden correcto, según la integridad referencial) son:

```
INSERT INTO Restaurantes VALUES (1234,'pizzahud','abascal 45','12345',2.0);
INSERT INTO "Áreas Cobertura" VALUES (1234,'12345');
INSERT INTO Horarios VALUES (1234,'X','12:00',to_date('23:00','HH24:MI'));
INSERT INTO Platos VALUES (1234,'pizza arrabiata',17.50,'pizza de carne y guindilla','picante');
INSERT INTO Clientes VALUES
('12345678N','Pedro','Pérez','Torralba',29,'4B','Madrid','12345','12345612','pedro','pedro');
INSERT INTO Descuentos VALUES (1100,to_date('20-04-09','HH24:MI'),50);
INSERT INTO Pedidos VALUES (1,'REST',to_date('17-02-09:19:50','DD-MM-YY:HH24:MI'),to_date('17-02-09:20:50','DD-MM-YY:HH24:MI'),34.25,'12345678N',1100);
INSERT INTO Contiene VALUES (1234,'pizza arrabiata',1,NULL,2);
```

Nota: La función **to\_date** permite convertir a tipo DATE y especificar el formato con el que se incluyen las fechas, con o sin horas (las fechas se almacenan internamente como valor numérico)

Nota: Se puede consultar las filas de las tablas con `SELECT * FROM nombreTabla;`

## Apartado 3: Carga de datos con SQL Loader

Borrar todas las filas de las tablas en el orden adecuado (tener en cuenta la integridad referencial) con `DELETE FROM nombreTabla;`

Importar con Oracle Loader las siguientes tablas (almacenadas en archivos de texto que se pueden descargar desde el campus virtual, con el mismo nombre de la tabla y extensión .txt). Generar los correspondientes archivos de control para realizar las importaciones. Hay que tener en cuenta las particularidades del formato DATE y que hay que usar una secuencia para generar el identificador de los pedidos. El esquema de un archivo de control es:

```
LOAD DATA
INFILE 'nombre_del_fichero'
APPEND
INTO TABLE nombre_de_la_tabla
FIELDS TERMINATED BY ';'
(
  primera_columna,
  ...,
  última_columna )
```

Examinar el archivo de informe generado como resultado de cada importación. ¿Qué errores de importación y qué restricciones de integridad se han violado? Para resolver los conflictos que surjan:

- Tener en cuenta que el código de restaurante 5677 es incorrecto (debería ser 5678).
- Sustituir “vegetariano” por “vegetal” y “torta carne especial” por “torta carne esp”.
- Ningún restaurante tiene horario partido (para cada horario repetido, dejar sólo la primera fila que aparezca).
- Eliminar los descuentos no válidos y los que no se puedan aplicar.

**Ayuda:** Se puede consultar el prontuario de Oracle disponible en el Campus Virtual, revisar la documentación electrónica del producto Utilities, Part II SQL\*Loader o la página web: [http://www.oraFAQ.com/wiki/SQL\\*Loader\\_FAQ](http://www.oraFAQ.com/wiki/SQL*Loader_FAQ)

Instrucción para lanzar el sql loader, desde una consola normal de línea de comandos del sistema operativo:

```
sqlldr userid= GIICxx @ BDiC / GIICxx control=restaurantes.ctl  
log=informe_restaurantes.txt
```

Ejemplos de ficheros de control:

```
LOAD DATA  
INFILE 'areas.txt'  
APPEND  
INTO TABLE "Areas Cobertura"  
  FIELDS TERMINATED BY ';' (restaurante,"codigo postal")
```

```
LOAD DATA  
INFILE 'pedidos.txt'  
APPEND  
INTO TABLE Pedidos  
  FIELDS TERMINATED BY ';' TRAILING NULLCOLS  
  (  
    "código" "Seq_CodPedidos.nextval",  
    estado,  
    fecha_hora_pedido "to_date(:fecha_hora_pedido,'DD-MM-YY:HH24:MI')",  
    fecha_hora_entrega "to_date(:fecha_hora_entrega,'DD-MM-YY:HH24:MI')",  
    "importe total",  
    cliente  
  )
```

## Apartado 4: Consultas SQL en el archivo `consultas.sql`

Codifica las consultas que se plantean, contestando en su caso las preguntas añadidas. Para ello incluye comentarios dentro del archivo `consultas.sql`:

1. Listado con todos los datos de los clientes, ordenados por apellidos.
2. Horarios de cada uno de los restaurantes. Para cada restaurante aparecerá su nombre y el día de la semana (sustituyendo la letra por el nombre completo del día) y la hora de apertura y de cierre, en formato HH:MM
3. Qué clientes (DNI, nombre y apellidos) han pedido alguna vez platos de la categoría “picante”?
4. ¿Qué clientes (DNI, nombre y apellidos) han pedido platos en todos los restaurantes?
5. ¿Qué clientes (DNI, nombre y apellidos) no han recibido aún sus pedidos?
6. Muestra todos los datos (salvo los platos que lo componen) del pedido (o pedidos) de mayor importe total. Considera que puede haber varios pedidos con el mismo importe.
7. Obtén el valor medio de los pedidos de cada cliente, mostrando su DNI, nombre y apellidos.
8. Muestra para cada restaurante (código y nombre) el número total de platos vendidos y el precio acumulado que obtuvieron.
9. Nombre y apellidos de aquellos clientes que pidieron platos de más de 15 €.
10. Para cada cliente (mostrar DNI, nombre y apellidos) mostrar el número de restaurantes que cubren el área en el que vive el cliente. Si algún cliente no está cubierto por ninguno, debe aparecer 0.